

## R-2.4

هذا الكود يحتوي على استدعاء دالة charge داخل نفسها بشكل متكرر دون وجود شرط أو حالة أساسية لإنهاء الاستدعاء المتكرر لأن هذا قد يؤدي إلى حدوث حلقة لا نهائية ونقص للفرامة لا ينتهي.

## R-2.5

أولاً: سيتم استدعاء دالة (charge) في class ال Credit card الأساسي بحيث ترجع هذه الدالة False إذا كان  $\text{blancet} + \text{Price} > \text{Limit}$  و true إذا كان لم يتحقق الشرط ثم تسند القيمة False أو true إلى متغير من نوع boolean (isSuccess) ثانياً: يتحقق الشرط إذا كانت قيمة isSuccess هي False في هذه الحالة سيقرر بفرمة الغرامة بشكل غير صحيح ←  $\text{super.charge}(5)$  لأن هذا الاستدعاء ليس له تأثير أو تعديل على الرصيد الحالي

## R-2.6

Fibonacci Progression with start values 2 and 2:

2 2 4 6 10 16 26 42

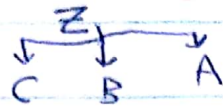
→ R 2.13

لا يمكن تحويل المتغير d الذي يشير إلى كائن من النوع Equestrian إلى متغير d من النوع Racer مباشرة لكن يجب علينا تحويل d أولاً من Equestrian إلى Horse ثم تحويل Horse إلى Racer كالتالي

Horse d = new Equestrian();

Racer r = (Racer) d;





R-2.10

عقد إجر أو تغيير في السؤال في الكلاس الأساسي (Z) يكون من الصعب تطبيق كل التعديلات في كل كلاس بشكل منفصل مما يؤدي إلى زيادة وجود أخطاء. صعوبة في الصيانة. من الصعب توفير الخصائص التي قد تحتاجها الكلاسات المستمدة من كلاسات أخرى

R-2.11

Read it.

Ship it.

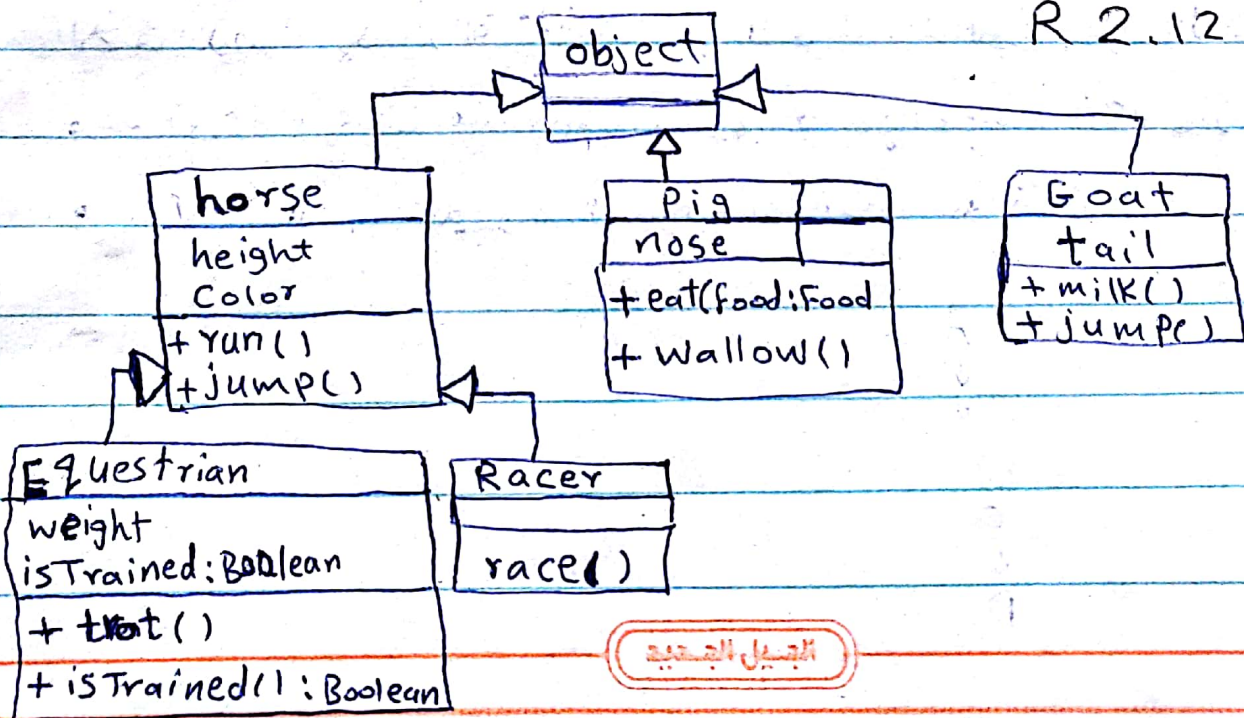
Buy it.

Read it.

Box it?

Read it.

R 2.12



الجيل الجديد

R-2.7

مدى البيانات من نوع long هو

- 9223372036854775808

+ 9223372036854775807

يمكن زيادة 128 في كل مرة، إلى ما يقارب 34 279 379 049 575 720 مرة قبل أن نتجاوز المدى لل long الذي يتصلها.

R-2.8

لا يمكن أن نوجهين أن تمررت من بعضهما لأنها تكونان دائرة ولا معنى ولا هدف منها

R-2.9

عند استدعاء دالة من class مشتق فيجب على compiler تتبع سلسلة التوابع الحقيقية للوصول إلى الدالة المناسبة فبذلك يزداد زمن البحث والوصول. وعند إنشاء كائن من class مشتق يجب بناء جميع الكلاسات الأساسية في سلسلة التوابع الحقيقية فبذلك يزداد زمن تهيئته وبناء الكائن

A  
↓  
B  
↓  
C  
↓  
D

الجيل الجديد